Learning With Physical Rounding for Linear and Quadratic Leakage Functions

Clément Hoffmann[1], Pierrick Méaux[2], Charles Momin[1], Yann Rotella[3], François-Xavier Standaert[1], Balazs Udvarhelyi[1]

[1] Crypto Group, ICTEAM Institute, UCLouvain, Louvain-la-Neuve, Belgium
[2] Luxembourg University, SnT, Luxembourg
[3] Université Paris-Saclay, UVSQ, CNRS, Laboratoire
de mathématiques de Versailles, 78000,Versailles, France

Abstract Fresh re-keying is a countermeasure against side-channel analysis where an ephemeral key is derived from a long-term key using a public random value. Popular instances of such schemes rely on key-homomorphic primitives, so that the re-keying process is easy to mask and the rest of the (e.g., block cipher) computations can run with cheaper countermeasures. The main requirement for these schemes to be secure is that the leakages of the ephemeral keys do not allow recovering the long-term key. The Learning with Physical Rounding (LWPR) problem formalizes this security in a practically-relevant model where the adversary can observe noise-free leakages. It can be viewed as a physical version of the Learning With Rounding (LWR) problem, where the rounding is performed by a leakage function and therefore does not have to be computed explicitly. In this paper, we first consolidate the intuition that LWPR cannot be secure in a serial implementation context without additional countermeasures (like shuffling), due to attacks exploiting worst-case leakages that can be mounted with practical data complexity. We then extend the understanding of LWPR in a parallel implementation setting. On the one hand, we generalize its robustness against cryptanalysis taking advantage of any (i.e., not only worst-case) leakage. A previous work claimed security in the specific context of a Hamming weight leakage function. We clarify necessary conditions to maintain this guarantee, based on the degree of the leakage function and the accuracy of its coefficients. On the other hand, we show that parallelism inherently provides good security against attacks exploiting worst-case leakages. We finally confirm the practical relevance of these findings by validating our assumptions experimentally for an exemplary implementation.

1 Introduction

Hard learning problems have been shown to be an important source of cryptographic hardness [Reg10,Pie12]. Popular instances of such problems include Learning Parity with Noise (LPN), Learning With Errors (LWE), which is a generalization to larger moduli [Reg05], and Learning With Rounding (LWR), which is a deterministic version [BPR12]. They have found many applications for the design of basic cryptographic primitives such as efficient authentication protocols [HB01,HKL+12] or pseudorandom functions [YS16], and advanced cryptographic primitives such as identity-based [GPV08,CHKP10] or fully homomorphic [Gen09,BV11] encryption. They have also gained particular interest in the context of post-quantum cryptography, as witnessed by the signature scheme CRYSTALS-Dilithium [DKL$^+$18] and the encryption scheme CRYSTALS-Kyber [BDK$^+$18], which have been selected by the NIST to become future standards.[1] Furthermore, their algebraic structure makes a (key-homomorphic) part of their implementation quite amenable to masking against side-channel attacks. Yet, protecting the error generation part remains a challenge, whether in the case of probabilistic (e.g., LPN-based) or deterministic (LWR-based) designs. This is because the noise generation itself can become the target of a side-channel attack [GLS14], and the rounding functions needed for the deterministic designs to be secure are non-linear and therefore more difficult to mask [BGL$^+$14].

Hard *physical* learning problems have been introduced as a possible remedy to these implementation issues. Conceptually, their goal is to leverage the physical features of an implementation in order to generate errors or to round, as required to implement primitives based on hard learning problems. A typical application of such physical problems is fresh re-keying against side-channel attacks [MSGR10], of which the goal is to generate an ephemeral (e.g., block cipher) key thanks to an easy-to-protect operation. Various pieces of work studied the cryptanalysis of such schemes instantiated with a binary field multiplication [BFG14,BCF$^+$15,PM16,GJ19]. Dziembowski et al. proved the security of binary re-keying based on a Learning Parity with Leakage (LPL) problem that reduces to the standard LPN problem [DFH$^+$16]. Unfortunately, the concrete level of noise in the leakages needed for their proof that LPL is secure was shown to be quite high. More recently, Duval et al. showed that it is possible to design re-keying schemes based on a Learning With Physical Rounding (LWPR) problem [DMMS21]. Here, the main observation is that many practical leakage functions, such as the Hamming weight one [MOP07], are non-injective. It is therefore tempting to let the leakage function perform the rounding, removing the hassle of computing this rounding explicitly and protecting it against side-channel attacks. Non-injectivity alone (i.e., without noise nor further constraints on the field in which computations take place) is known to be insufficient. For example, in binary fields the Hamming weight of a secret value provides a linear equation of its bits. But combining a non-injective leakage function with computations in a prime field was suggested as a possible source of "crypto-physical" hardness, emulating Boneh et al.'s cryptographic dark matter [BIP$^+$18].

It is easy to see that practically-relevant secure LWPR instances could be a game changer for the secure implementations of cryptographic algorithms. First, masked block ciphers usually incur performance overheads that are quadratic in the number of shares [ISW03,GR17]. By contrast, the cost of masked key-homomorphic primitives scales linearly in this number and can benefit from simple refreshing schemes (with linear randomness requirements) [BDF$^+$17]. Second, the analysis of key-homomorphic primitives in the probing model is trivial and does not raise composability issues since, as for linear functions, their com-

[1] `https://csrc.nist.gov/Projects/post-quantum-cryptography`.

putation can be implemented share by share and is therefore probe-isolating non-interferent [CS20]. Third, they intrinsically mitigate the risks of security flaws caused by physical defaults like glitches [MPG05] or transitions [CGP$^+$12] that can lead to shares re-combinations, thanks to the independent manipulation of the shares they allow [CS21]. Eventually, is was recently hinted that they also have good potential to significantly improve the performance vs. side-channel security tradeoff of CCA-secure public key encryption schemes [HLM$^+$23]. The latter is well-motivated given the acknowledged difficulty to protect schemes like CRYSTALS-Kyber against side-channel attacks [RRCB20,NDGJ21,UXT$^+$22].

All these reasons give a strong incentive for understanding the security of the LWPR assumption under realistic leakage functions. The CHES 2021 work of Duval et al. made a first step in this direction by analyzing LWPR in the meaningful though specific case of Hamming weight leakages, and proposed parallel (FPGA) implementations processing 124 bits per cycle as a first cryptanalysis target. This investigation therefore suggested as fundamental problem the analysis of leakage functions that (even slightly) deviate from the Hamming weight case, while also leaving doubts regarding the possibility to ensure the security of LWPR in a serial (e.g., 32-bit software) implementation context.

In this paper, we contribute to these problems in three main directions.

As an appetizer, we confirm the intuition that securing LWPR in a serial implementation context requires additional side-channel countermeasures. This is because attacks exploiting so-called worst-case leakages can then be mounted with practical data complexity. By worst-case leakages, we mean the most informative values provided by the leakage function (e.g., the extreme ones in the Hamming weight case). This observation suggests the use of shuffling as a natural complement to serial LWPR which, if well implemented, has the effect of emulating a parallel implementation [HOM06,VMKS12,UBS21]. It also allows us to focus the rest of our investigations on parallel LWPR instances.

Next, we generalize the security of the LWPR assumption beyond the Hamming weight function. For this purpose, we first consider attacks taking advantage of any value of the leakage function, which we denote as "any-case leakages" in the paper. We show that security against algebraic cryptanalysis is preserved in the practical context of linear leakage functions (i.e., functions that can be expressed as a weighted sum of the bits manipulated by an implementation) [SLP05], under reasonable conditions on the accuracy of the weights. We also argue that these guarantees can be extended towards higher-order (e.g., quadratic) leakage functions. We then show that attacks exloiting worst-case leakages (used to rule out unprotected serial LWPR instances) can be prevented in the parallel case, with similar assumptions as to resist any-case leakages.

We combine these investigations with experiments confirming the practical relevance of our assumptions for realistic hardware implementations. As a result, we make a crucial step towards making deterministic hard physical learning problems a credible alternative to secure cryptographic implementations. And we conclude the paper by identifying interesting scopes for further research.

We note that the security of LWPR with linear or quadratic leakage functions relies on different arguments than previously used for similar constructions. For example in [DMMS21], the high-level idea behind the hardness of LWPR with Hamming weight leakages is that the composition of linear functions in different structures is sufficient to provide security. This holds since the composition gives functions with good cryptographic parameters in both initial structures. It corresponds to the main conclusion of Boneh et al.'s cryptographic dark matter [BIP$^+$18] which proposes PRF constructions by only combining additions modulus p and modulus q for any pair of different primes p and q. The same general idea has been reused recently in [DGH$^+$21]. In the case of LWPR with linear or quadratic leakage functions, additionally to the composition of functions over two different structures, good cryptographic properties are obtained directly from the non-injectivity of one of the functions. This new insight allows us to generalize the results to more functions and to strengthen their connection to practice, since the non-injectivity of a leakage function is typically something that can be characterized by existing side-channel evaluation setups & tools.

2 Background

2.1 Learning With physical rounding

We first define the LWPR assumption introduced in [DMMS21]. For this purpose, let us consider a secret matrix $\boldsymbol{K} \in \mathbb{F}_2^{m\times n}$ and a public vector $\boldsymbol{r} \in \mathbb{F}_2^n$. The work of Boneh et al. on cryptographic dark matter showed that a low-complexity wPRF $\mathsf{F}_{\boldsymbol{K}}(\boldsymbol{r})$ can be obtained by computing the product $\boldsymbol{K}\cdot\boldsymbol{r}$, interpreting its output as a vector of 0/1 values over $\mathbb{F}_3$ and rounding it. Precisely:

$$\mathsf{F}_{\boldsymbol{K}}(\boldsymbol{r}) := \mathsf{map}(\boldsymbol{K}\cdot\boldsymbol{r}),$$

where the proposed mapping was the sum of this vector of 0/1 values modulo 3 [BIP$^+$18]. The idea of Duval et al. is to replace this mathematical mapping by a physical one (the leakage function) that is formalized as follows.

First, it is assumed that the leaking device computes on binary-represented data: each value in $\mathbb{F}_p$ is therefore represented with (at least) $\lceil \log p \rceil$ bits. We denote as $\mathsf{g} : \mathbb{F}_p \to \{0,1\}^{\lceil \log p \rceil}$ the function associating to each element y of $\mathbb{F}_p$ the binary representation of its representative in $[0, p-1]$. We also define $\mathsf{g}_m : \mathbb{F}_p^m \to \{0,1\}^{m\lceil \log p \rceil}$ as: $\mathsf{g}_m(\boldsymbol{y}) := \mathsf{g}(\boldsymbol{y}_1)||\mathsf{g}(\boldsymbol{y}_2)||\ldots||\mathsf{g}(\boldsymbol{y}_m)$. This assumption is quite generic and captures the reality of most embedded computing devices deployed in current applications. Next, a more specialized assumption is required to define how the physical (noise-free) leakages depend on the $m\lceil \log p \rceil$ bits provided by g_m. This role will be played by the leakage function. We denote the leakage function computed on the binary representation of the manipulated data as $\mathsf{L}_\mathsf{g}(.)$. The generic LWPR problem is then defined as follows:

Definition 1 (Learning with physical rounding). *Let $p, n, m \in \mathbb{N}^*$, p prime, for $\boldsymbol{K} \in \mathbb{F}_p^{m\times(n+1)}$. The $\mathrm{LWPR}_{\mathsf{L}_\mathsf{g},p}^{n,m}$ sample distribution is given by: $\mathcal{D}_{\mathrm{LWPR}_{\mathsf{L}_\mathsf{g},p}^{n,m}} :=$*

$\big(\boldsymbol{r}, \mathsf{L}_{\mathsf{g}}(\boldsymbol{K} \boxdot \boldsymbol{r})\big)$ *for* $\boldsymbol{r} \in \mathbb{F}_p^n$ *uniformly random, where* $\boldsymbol{K} \boxdot \boldsymbol{r} = \boldsymbol{K} \cdot (\boldsymbol{r}, 1)$ *and* $\mathsf{L}_{\mathsf{g}} : \mathbb{F}_p^m \to \mathbb{R}^{n_d}$ *is the physical rounding function. Given query access to* $\mathcal{D}_{\mathrm{LWPR}^{n,m}_{\mathsf{L}_{\mathsf{g}},p}}$ *for a uniformly random* $\boldsymbol{K}$*, the* $\mathrm{LWPR}^{n,m}_{\mathsf{L}_{\mathsf{g}},p}$ *problem is* (q, τ, μ, ϵ)*-hard to solve if after the observation of* q *LWPR samples, no adversary can recover the key* $\boldsymbol{K}$ *with time complexity* τ*, memory complexity* μ *and probability higher than* ϵ*.*

Duval et al. analyzed the case of Hamming weight leakages as a first step. The Hamming weight function is defined on any vector $\boldsymbol{v}$ of length $t \in \mathbb{N}^*$ with coefficients in $\{0, 1\}$ as $\mathsf{M}_{\mathsf{h}}(\boldsymbol{v}) = \sum_{i=1}^{t} \boldsymbol{v}_i$, where the sum is performed in $\mathbb{Z}$. They additionally considered two implementation contexts. In the serial case, the adversary is provided with m leakages on $\lceil \log p \rceil$ bits:

$$\mathsf{M}_{\mathsf{h}}^{s}(\boldsymbol{y}) : \boldsymbol{y} \mapsto \Big(\mathsf{M}_{\mathsf{h}}\big(\mathsf{g}(\boldsymbol{y}_1)\big), \mathsf{M}_{\mathsf{h}}\big(\mathsf{g}(\boldsymbol{y}_2)\big), \ldots, \mathsf{M}_{\mathsf{h}}\big(\mathsf{g}(\boldsymbol{y}_m)\big)\Big).$$

In the parallel case, she receives the sum of these m values, denoted as $\mathsf{M}_{\mathsf{h}}^{p}(\boldsymbol{y})$.

The LWPR instance proposed by Duval et al. uses a Mersenne prime $p = 2^{31} - 1$. It aims at the generation of 124-bit fresh keys in the parallel case and therefore considers $m = 4$. As for the main security parameter n, their analysis suggested $(n + 1)\log(p) + 3\log(n) \geq 124$ as a lower bound. This led the authors to select $n = 4$ as a first target for further cryptanalytic investigations.

2.2 Regression-based side-channel analysis

Linear regression is among the most popular tools for profiling a side-channel leakage model. Originally introduced by Schindler et al. [SLP05], it has been rapidly established as an efficient alternative to template attacks [GLP06]. Its main idea is to approximate the deterministic part of the leakage function as a weighted sum of n_b well-chosen basis function that we denote as β_i:

$$\mathsf{M}_{\mathsf{r}}(\boldsymbol{v}) = \sum_{i=1}^{n_b} a_i \cdot \beta_i(\boldsymbol{v}),$$

with the a_i's $\in \mathbb{R}$. For a t-bit value $\boldsymbol{v}$, a typical choice is to consider $n_b = t$ and to use the bits of $\boldsymbol{v}$ as basis functions.[2] We will denote the resulting class of leakage functions as linear models $\mathsf{M}_{\mathsf{r1}}(\boldsymbol{v})$, which generalizes Hamming weight leakages where $a_i = 1 \; \forall \; 1 \leq i \leq n_b$. The model can be refined by adding more basis functions. Typically, we can add the $\binom{t}{\delta}$ basis functions of degree δ, which we denote as quadratic leakage models $\mathsf{M}_{\mathsf{r2}}(\boldsymbol{v})$ when $\delta = 2$, cubic leakage models $\mathsf{M}_{\mathsf{r3}}(\boldsymbol{v})$ when $\delta = 3$, ..., until the bijective model $\mathsf{M}_{\mathsf{r}t}$ where $\delta = t$ and $n_b = 2^t$.

Profiling a leakage model then essentially boils down to estimate the vector of coefficients $\boldsymbol{a}$, by applying the least square method. For this purpose, the evaluator first collects a vector of n_p profiling traces $\boldsymbol{l} = [l^1, l^2, \ldots, l^{n_p}]$, corresponding

[2] Or $n_b = t + 1$, with a constant term to capture DC effects in the measurements.

to values $\boldsymbol{v}^1, \boldsymbol{v}^2, \ldots, \boldsymbol{v}^{n_p}$. She then builds a $n_p \times n_b$ matrix:

$$B = \begin{pmatrix} \beta_1(\boldsymbol{v}^1) & \beta_2(\boldsymbol{v}^1) & \cdots & \beta_{n_b}(\boldsymbol{v}^1) \\ \beta_1(\boldsymbol{v}^2) & \beta_2(\boldsymbol{v}^2) & \cdots & \beta_{n_b}(\boldsymbol{v}^2) \\ \vdots & \vdots & \ddots & \vdots \\ \beta_1(\boldsymbol{v}^{n_p}) & \beta_2(\boldsymbol{v}^{n_p}) & \cdots & \beta_{n_b}(\boldsymbol{v}^{n_p}) \end{pmatrix}.$$

Eventually, the vector of coefficients minimizing the means square error is:

$$\hat{\boldsymbol{a}} = [\hat{a_1}, \hat{a_2}, \ldots, \hat{a_{n_b}}] = (B^T \cdot B)^{-1} \cdot B^T \cdot \boldsymbol{l}.$$

We note that the analysis in [SLP05] considers noisy leakages and therefore adds a probabilistic component to the previous deterministic function. We do not detail this part since LWPR (conservatively) assumes noise-free leakages.

2.3 Correlation-based security evaluations

Given a leakage model, the main question of a side-channel security evaluation is to determine the number of attack traces needed to recover a key [SMY09]. In the context of univariate leakage functions that we consider in this paper, Pearson's correlation is among the most popular evaluation tools [BCO04]. We next describe the basic approximations we will use in our empirical analyses.

Let us first denote a vector of n_a attack values $\overline{\boldsymbol{v}} = [\boldsymbol{v}^1, \boldsymbol{v}^2, \ldots, \boldsymbol{v}^{n_a}]$. From those values, an evaluator can compute the modeled leakage vector:

$$\boldsymbol{m} = [\mathsf{M_r}(\boldsymbol{v}^1), \mathsf{M_r}(\boldsymbol{v}^2), \ldots, \mathsf{M_r}(\boldsymbol{v}^{n_a})].$$

Similarly, she can measure a vector of actual leakage traces $\boldsymbol{l} = [l^1, l^2, \ldots, l^{n_a}]$. Viewing these vectors as n_a samples of random variables M and L, the data complexity of a Correlation Power Analysis (CPA) is given by [Man04]:

$$N = \frac{c}{\hat{\rho}(M, L)^2},$$

where c is a small constant and $\hat{\rho}$ denotes Pearson's correlation coefficient.

In our experimental evaluations, we will in particular be interested by the comparison between different leakage models, with the goal to evaluate whether simplifying a model (e.g., by limiting its degree or quantizing it) results in a significant information loss. Pearson's correlation is convenient for this purpose. For example, let us assume a model M_1 that is a simplification of a model M_2, leading to model errors captured by a random variable E so that $M_1 = M_2 + E$. Then, thanks to the correlation chain rule given in [SPRQ06] we have:

$$\hat{\rho}(M_1, L) = \hat{\rho}(M_1, M_2) \cdot \hat{\rho}(M_2, L).$$

As a result, the increase of data complexity that is due to simplifying the model M_2 into M_1 is reflected by the factor $f = \frac{1}{\hat{\rho}(M_1, M_2)^2}$: if $f \approx 1$ the attack using the simplified model is close to the one using the complex model; if $f > 1$ the simpler model misses some leakage features and the attack using this model requires f times more traces than the attack using the more complex model.

2.4 Cryptographic criteria & p-ary functions

Since the re-keying scheme we consider in this paper is linear over $\mathbb{F}_p$, we will study the properties of the leakage function embedded in $\mathbb{F}_p$ that a side-channel adversary can obtain. For this purpose, we adapt tools from the analysis of cryptographic criteria for Boolean functions from $\mathbb{F}_2^n$ to $\mathbb{F}_2$ (e.g., [Car21]) and study the cryptographic properties of functions from $\mathbb{F}_p^n$ to $\mathbb{F}_p$.

Definition 2 (p-ary function). *For p a prime, a p-ary function f in n variables (an n-variable p-ary function) is a function from $\mathbb{F}_p^n$ to $\mathbb{F}_p$. The set of all p-ary functions in n variables is denoted by $\mathcal{F}_{p,n}$, and $|\mathcal{F}_{p,n}| = p^{p^n}$.*

Definition 3 (Algebraic normal form and degree (e.g., [Hou18])). *We call Algebraic Normal Form (ANF) of a p-ary function f its n-variable polynomial representation over $\mathbb{F}_p$ belonging to $\mathbb{F}_p[x_1, \ldots, x_n]/(x_1^p - x_1, \ldots, x_n^p - x_n)$:*

$$f(x) = \sum_{S \subset [0,p-1]^n} a_S \left(\prod_{i \in [1,n]} x_i^{S_i} \right) = \sum_{S \subset [0,p-1]^n} a_S x^S,$$

where $a_S \in \mathbb{F}_p$. The ANF of f is unique, and the algebraic degree of f is defined as $\mathsf{deg}(f) = \max_{\{S \mid a_S \neq 0\}} \sum_{i=1}^{n} S_i$ (with the convention that $\mathsf{deg}(0) = 0$).

Definition 4 (Nonlinearity). *The nonlinearity $\mathsf{nl}(f)$ of a p-ary function $f \in \mathcal{F}_{p,n}$, is the minimum Hamming distance between f and all the affine functions in $\mathcal{F}_{p,n}$:*

$$\mathsf{nl}(f) = \min_{g,\, \mathsf{deg}(g) \leq 1} \{d_H(f,g)\},$$

where $d_H(f,g)$ is the Hamming distance $|\{x \in \mathbb{F}_p^n \mid f(x) \neq g(x)\}|$ between f and g.

3 Appetizer: threat model and serial LWPR

Before describing our technical contributions, we quickly recall the threat model we consider, which is illustrated in Figure 1. We also come back on the impact of an attack against serial LWPR instances outlined in the appendix of [DMMS21], by specializing it to the case of a Hamming weight function for simplicity. We use this example for two purposes: first, confirming that securing serial LWPR instances requires additional countermeasures; second: introducing the difference between attacks using worst-case leakages and any-case leakages.

In brief, the idea of fresh re-keying schemes is that it is possible to efficiently mask their key-homomorphic part (i.e., the product between the secret matrix $\boldsymbol{K}$ and the public vector $\boldsymbol{r}$) and to recombine the output of this product for which the adversary can only observe the leakage. In other words, the multiplication we consider in Definition 1 will be performed on the shares $\boldsymbol{K}^1, \boldsymbol{K}^2, \ldots, \boldsymbol{K}^q$ and then recombined such that the fresh key is $\boldsymbol{K} \boxdot \boldsymbol{r} = \sum_{i=1}^{q} \left(\boldsymbol{K}^i \boxdot \boldsymbol{r}\right)$.

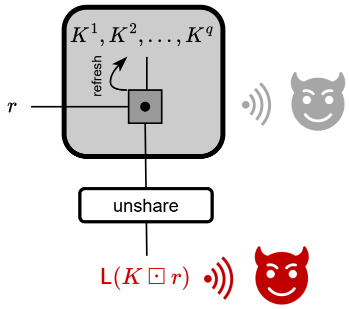

Figure 1: Fresh re-keying with LWPR: threat model.

This fresh key will then be used without masking for a single (e.g., block cipher) encryption, which is expected to be significantly harder to break than if using the long-term key multiple times with the same cipher. As a result, there are two natural attack paths against such re-keying schemes. The first one (depicted in gray on Figure 1) is to target the masked product. Evaluating it can be done with standard side-channel analysis techniques, enjoying the advantages listed in introduction (linear overheads, no composability issues, resistance against glitches). The analysis of this attack path given in [DMMS21] remains identical and we therefore do not repeat it. The second option (depicted in red on Figure 1) is to target the output of the product after its shares have been re-combined. Evaluating it requires less standard techniques and can be done in different models. Binary fresh re-keying schemes like [MSGR10] require noisy leakages to be secure. We consider a more conservative model where the adversary obtains noise-free leakages. As a result, the analysis of this attack path becomes conceptually similar to the analysis of a stream cipher, the product playing the role of an LFSR and the leakage function the one of a filter.

Given this threat model, the attack against serial LWPR implementations is pretty simple. Say we have an instance of LWPR with $p = 2^{31} - 1$ where the adversary can observe the Hamming weight of every 31-bit word of $\boldsymbol{K} \boxdot \boldsymbol{r}$. Then, she can filter the leakages and retain the ones with (worst-case) value 0, which has a single preimage. Every such event gives a linear equation in the key elements of the corresponding line of $\boldsymbol{K}$, and each such line can be recovered with $(n+1)$ linear equations. Since p is only mid-size, these events happen with a concretely reachable probability $\frac{1}{p}$. As a result, after the generation $p\,(n+1)$ LWPR samples, the full key is compromised with good probability.

Since it is hard to rule out that (close to) Hamming weight leakages can be observed in practice (as Section 6 will confirm experimentally), this attack suggests that serial instances of LWPR cannot be secure without additional side-channel countermeasures. Various candidates can be considered for this purpose. As it is argued in [DMMS21], exploiting extreme Hamming weights becomes difficult when the level of parallelism increases (since they become exponentially less likely), and shuffling therefore appears as the most natural one. Indeed, if well implemented, shuffling makes the m intermediate computations of a serial implementation hard to distinguish so that the adversary does not gain more information than if she was provided with the sum of these leakages, which emulates a parallel implementation [HOM06,VMKS12,UBS21].[3] We show in Section 5 that attacks exploiting worst-case leakages can be generalized, and that a security argument can be given based on the degree of the (linear) leakage function and the accuracy of its coefficients. Beforehand, we consider another important class of (algebraic) attacks where the adversary exploits any-case leakages (i.e., where the leakages are not filtered and all the measurements are used). For this purpose, we generalize the former security analysis of Duval et al. from the specialized Hamming weight leakage function towards any linear leakage function. Interestingly, for those attacks we are even able to give a security argument for serial implementations, which then easily extends to parallel ones.

4 LWPR for linear leakage functions

While reasonable as a first step, considering security against Hamming weight leakages is oversimplifying since the different bits of an implementation can consume more or less power, due to different load capacitances. This is precisely what is modeled by regression-based attacks [SLP05]. In this section, we therefore analyze the security of LWPR in this generalized context.

For this purpose, we first observe that the product $\boldsymbol{K} \boxdot \boldsymbol{r}$ is linear over $\mathbb{F}_p$. As a result, we focus on the (linear invariant) cryptographic criteria of the remaining function L_g considered as a function over $\mathbb{F}_p$. If the adversary can approximate well the real function L_g by a p-ary function, she obtains a system of equations over $\mathbb{F}_p$ where the unknowns are (affine combinations of) the key elements.

This gives a modeling of $\mathsf{L}_\mathsf{g}(\boldsymbol{K} \boxdot \boldsymbol{r})$ as equations over $\mathbb{F}_p$ where the only unknowns are the $\mathbb{F}_p$ elements of $\boldsymbol{K}$. We next carry out such an analysis for a linear leakage model in the serial case, by considering these models as functions over $\mathbb{F}_p$, which we define as s-bounded pseudo-linear functions.

Definition 5 (s-bounded pseudo-linear functions). *We denote as $\mathsf{F}_{\boldsymbol{a}}$ the function characterized by $\boldsymbol{a} \in \mathbb{F}_p^t$, where $t = \lceil \log p \rceil$ and defined as:*

$$\mathsf{F}_{\boldsymbol{a}} : \ \mathbb{F}_p \to \mathbb{F}_p, \ y \mapsto \sum_{i=0}^{t-1} a_i \cdot \mathsf{g}(y)_i. \tag{1}$$

We call $\mathsf{F}_{\boldsymbol{a}}$ s-bounded pseudo-linear if each a_i belongs to $[0, s]$, with $s \in \mathbb{F}_p$ a security parameter, and we call C_1^s the class of s-bounded pseudo-linear functions.

We recall that the g function associates elements of $\mathbb{F}_p$ to their binary representation. The name pseudo-linear comes from the fact that this function is linear in the bits of the binary representation of $\boldsymbol{a}$. However, it corresponds to a much

[3] Increasing the size of p could provide similar security benefits at higher cost.

higher degree p-ary function. This increase of the degree when projecting a binary representation into a prime field is the root of the LWPR hardness. We next consider that an s-bounded $\mathsf{F}_{\boldsymbol{a}}$ is the p-ary function used to approximate L_{g}.

We note that this definition implicitly relies on two assumptions: one on the degree of the physical leakage function and another one on the number of values taken by the coefficients a_i. The practical relevance of these two assumptions is discussed in Section 6. We also note that the proposed analysis captures an important class of (algebraic) attacks against LWPR. We will show that the non-injectivity of a one-variable p-ary function is sufficient to prove a lower bound on its degree and its nonlinearity. On the one hand, the high algebraic degree of a p-ary function allows thwarting the attacks based on solving a linearized algebraic system. On the other hand, the high nonlinearity prevents the attacks based on solving noisy linear algebraic systems, which use the best approximation of a p-ary function by an affine function. But as usual in cryptanalysis, nothing prevents that other characterizations or attack vectors lead to better results.

Definition 6 (Preimage sets & main preimage). *Let $f \in \mathcal{F}_{p,m}$, we call:*

- $\mathcal{A}_f(y) = \{x \in \mathbb{F}_p^m \mid f(x) = y\}$ *for* $y \in \mathbb{F}_p$*, the set of preimages of* y *through* f*.*
- $\mathsf{u}_f \in \mathbb{F}_p$ *a main image of* f *defined as* $\forall y \in \mathbb{F}_p, |\mathcal{A}_f(\mathsf{u}_f)| \geq |\mathcal{A}_f(y)|$*.*
- $\mathsf{v}_f \in [1, p^m]$ *the main preimage size of* f*, defined as* $\mathsf{v}_f = |\mathcal{A}_f(\mathsf{u}_f)|$*.*
- $\mathsf{w}_f \in [0, p-1]$ *the no preimage set's size of* f*, defined as*

$$\mathsf{w}_f = |\{y \in \mathbb{F}_p \mid \mathcal{A}_f(y) = \emptyset\}|$$

We first show a bound on the degree of f based on its main preimage size.

Proposition 1 (Main preimage size and algebraic degree). *Let $f \in \mathcal{F}_{p,1}$, if its main preimage $\mathsf{v}_f < p$ then its algebraic degree $\mathsf{deg}(f) \geq \mathsf{v}_f$.*

Proof. Since $\mathsf{v}_f < p$, f is not a constant function and $f - \mathsf{u}_f$ is not the null function. Accordingly, the function $f - \mathsf{u}_f$ admits v_f different zeros in $\mathbb{F}_p$, and there exists v_f terms $(x - z_i)$ dividing f. Therefore its degree is at least v_f and, since the algebraic degree is affine invariant, $\mathsf{deg}(f) = \mathsf{deg}(f - \mathsf{u}_f) \geq \mathsf{v}_f$. □

Furthermore, when f is not enough non-injective, composing the function can be used to determine better its degree as shown in the following corollary:

Corollary 1. *Let $f \in \mathcal{F}_{p,1}$ and $r \in \mathbb{N}^*$, we denote $f^{\circ r} = f \circ f \circ \cdots \circ f$ the composition of f r times. If $\mathsf{v}_{f^{\circ r}} < p$ then $\mathsf{deg}(f) \geq (\mathsf{v}_{f^{\circ r}})^{1/r}$.*

We next show a bound on the nonlinearity of f.

Proposition 2 (Main preimage size, no preimage set's size and nonlinearity). *Let $f \in \mathcal{F}_{p,1}$ such that its main preimage size $\mathsf{v}_f < p$, then:*

$$\mathsf{nl}(f) \geq \min\left(p - \mathsf{v}_f, \max\left(\mathsf{v}_f - 1, \mathsf{w}_f\right)\right).$$

Proof. We derive the bound on the nonlinearity by first considering the distance to constant functions, and then to any affine function. Since $\mathsf{v}_f < p$, f is not constant, and it will agree on at most v_f values with a constant function c. Hence $d_H(f, c) \geq p - \mathsf{v}_f$. Besides, p-ary affine (not constant) functions in one variable are bijective. Accordingly, each element of $\mathbb{F}_p$ is the image of exactly one element. Therefore, for any affine (not constant) function in $\mathcal{F}_{p,1}$, there exists at least $\mathsf{v}_f - 1$ elements of $\mathbb{F}_p$ where the image of f is different from the affine function's one (all except one of the preimages of u_f). Moreover, since w_f elements are not in the image of f, f disagrees at least w_f times with any affine (not constant) function, which give the final bound. □

Finally, we bound the main preimage size of s-bounded pseudo-linear functions.

Proposition 3 (Properties of s-bounded pseudo-linear functions). *Let $f \in \mathsf{C}_1^s$ with $ts < p$, where $t = \lceil \log p \rceil$, then the following holds:*

- $\mathsf{v}_f \geq \lceil \frac{p}{ts+1} \rceil$,
- $\mathsf{w}_f \geq p - ts - 1$.

And assuming $\mathsf{v}_f \neq p$, we further have:

- $\mathsf{deg}(f) \geq \lceil \frac{p}{ts+1} \rceil$,
- $\mathsf{nl}(f) \geq \min \left(p - \mathsf{v}_f, \max \left(\lceil \frac{p}{ts+1} \rceil - 1, p - ts - 1 \right) \right)$.

Proof. Since $ts < p$, we obtain that for all $x \in \mathbb{F}_p$, $0 \leq f(x) \leq ts$, hence $\mathsf{w}_f \geq p - ts - 1$. Thereafter, $f(x)$ can take at most $ts + 1$ values, hence there exists at least a preimage set with cardinal $\lceil \frac{p}{ts+1} \rceil$ or more. The last two items are obtained using Proposition 1, Proposition 2 and two first items. □

Note that the condition $ts < p$ guarantees that no reduction takes place independent of the input of the pseudo-linear leakage function in Equation 1.

For a given $\boldsymbol{a} \in \mathbb{F}_p^t$, the output of $\mathsf{F}_{\boldsymbol{a}}$ gives a system of equations where the key elements are the unknowns. Therefore, solving the algebraic system given by images of $\mathsf{F}_{\boldsymbol{a}}$ allows retrieving the key. We next explore two methods for solving such a system over $\mathbb{F}_p$, and demonstrate that the time and data complexity of these methods are sufficiently high for our instances in Section 6.5 .

Concretely, an adversary can either solve the system of k variables directly or use the higher-order correlation approach presented in [Cou02]. Higher-order correlation attacks consist in approximating $\mathsf{F}_{\boldsymbol{a}}$ by a degree-d function h, and then solving systems of equations until one is such that $\mathsf{F}_{\boldsymbol{a}}$ and h coincide on all these equations. An adversary can create additional equations if needed, for instance through a linear combination of the existing ones. Hence, the time complexity of solving a noisy degree-d system of equations over $\mathbb{F}_p$ can be written as $C(1-\varepsilon)^{-D}$ with C the time complexity to solve a degree-d system of equations in k' variables over $\mathbb{F}_p$, $(1-\varepsilon)$ the probability of the approximation to be correct for one equation, k' the number of variables (i.e., at least the number of key variables k, but it can be more if techniques introducing new variables are used), and D the quantity of data necessary (at least the number of variables k').

Exact algebraic system attack path. Let us for now fix the security parameter s to 2^{12} (the practical relevance of this value will be discussed in Section 6). Trying to solve the system directly given by the output of the function $\mathsf{F}_{\boldsymbol{a}}$ with this value and a modulus $p = 2^{31} - 1$, we find $\mathsf{v}_f \geq \lceil \frac{2^{31}-1}{31 \cdot 2^{12}+1} \rceil \geq 2^{14}$ (since $t = \lceil \log(p) \rceil$). Therefore, the adversary would have to solve a system of equations of degree at least 2^{14} in $n+1$ variables over $\mathbb{F}_{2^{31}-1}$ to recover the first row of $\boldsymbol{K}$. There are usually two main families of algorithms considered to solve polynomial systems of equations: the XL algorithms and Gröbner bases algorithms such as F4 [Fau99] or F5 [Fau02]. The complexities of these families of algorithms are studied in different works. We refer to [YC04] for the XL family and to [BFS15] for F4/F5. For clarity, we first discuss the results given by the simpler approach of linearization. Let V_d denote the number of monomials over $\mathbb{F}_p$ with degree at most d in the k variables of the system. The linearization attacks consists in considering each of these monomials as a new variable, and then in inverting the linear system using Gaussian elimination. Thereafter, the corresponding time complexity to solve a system with this approach is simply $\mathcal{O}\left(V_d^3\right)$. XL algorithms or Gröbner bases algorithms should improve over this complexity. However, Gaussian elimination algorithms give a convenient estimate of the complexity to solve a linear system, that is sufficient for our purposes and which can be combined with mild security margins if needed. For example, XL and F4/F5 can be much faster when more polynomials are available, leading to a time complexity reduced to $\mathcal{O}\left(V_d^{\omega}\right)$, where $2 \leq \omega \leq 3$ is a linear algebra constant that depends on the algorithm used. We make the conservative choice to consider the complexity $\mathcal{O}\left(V_d^2\right)$ for our concrete security estimations.

Noisy linear system attack path. We also consider the complexity of linearizing the d-degree system, then solving the resulting linear system. For this purpose, we use a similar argument as the one given in [DMMS21]. Namely, we first observe that the number of variables after linearization is $V_d = |\{\boldsymbol{v} \in [0, p-1]^k,\ 0 \leq \sum_{i=1}^{k} \boldsymbol{v}_i \leq d\}| = \sum_{l=0}^{d} \binom{l+k-1}{l} = \binom{k+d}{k}$ (using the stars and bars theorem) and again assume $C = \mathcal{O}\left(V_d^{\omega}\right)$. The noisy linear system approach then just implies considering a probability of error $\varepsilon = \mathsf{nl}(\mathsf{F}_{\boldsymbol{a}})/p$ for the equation approximation. In this case, with a degree $d = 1$, we have $V_1 = k+1$ and the time complexity is at least $\mathcal{O}\left((k+1)^{\omega}(p/(p - \mathsf{nl}(\mathsf{F}_{\boldsymbol{a}})))^k\right)$. Using the bound of Proposition 3 therefore provides the required security estimation.

These theoretical bounds will be made concrete in Section 6.5, where actual instances of $\mathsf{F}_{\boldsymbol{a}}$ are given, with their associated values v_f and w_f.

Adaptation to the parallel case. In the serial case above, we studied the properties of the function an adversary can obtain from the leakage corresponding to one ($\mathbb{F}_p$) element of the ephemeral key, each element of $\boldsymbol{y}$ depending on a different row of the long term key. In the parallel case, the adversary gets less information since she obtains the leakage on the whole $\boldsymbol{y}$ and not independently on each coefficient. In this case, the leakage is the sum of the leakages obtained in the serial case. Therefore, in terms of p-ary functions, calling f_i the functions considered in the serial case, the one in the parallel case is $f' \in \mathcal{F}_{p,m}$:

$$f'(\boldsymbol{y}) = \sum_{i=1}^{m} f_i(y_i).$$

Since f' is a direct sum of functions (sum of functions acting on different variables) its properties can be derived from the ones of the f_i's. Namely, $\mathsf{deg}(f_i)$ and $\mathsf{nl}(f_i)$ can be obtained from Proposition 3, $\mathsf{deg}(f') = \max_i \mathsf{deg}(f_i)$ and the nonlinearity of f' can be derived from the following result:

Property 1 (Adapted from [CHMS22], Proposition 1*).* Let $\mathbb{G}$ be a group, Let $h = f + g$ be the direct sum of the functions f and g with n and m $\mathbb{G}$-variables respectively. Then, $\mathsf{nl}_{\mathbb{G}}(h) \geq \max(|\mathbb{G}|^n \mathsf{nl}_{\mathbb{G}}(g), |\mathbb{G}|^m \mathsf{nl}_{\mathbb{G}}(f))$.

Accordingly, we also obtain $\mathsf{nl}(f') \geq \max_i p^{m-1}\mathsf{nl}(f_i)$. The complexity of the attacks against parallel LWPRG implementations is therefore increased thanks to this higher nonlinearity and the higher number of variables km.

5 Security against worst-case leakages

We now extend the approach of [DMMS21] for attacks taking advantage of worst-case leakages (intuitively outlined in Section 3). Precisely, we investigate the complexity of attacks that benefit from the existence of leakages with small preimage sets. In the parallel case, the adversary obtains images z such that:

$$z = f'(\boldsymbol{y}) = \sum_{i=1}^{m} f_i(y_i) = \sum_{i=1}^{m} f_i(\boldsymbol{k}_i \boxdot \boldsymbol{r}),$$

where $\boldsymbol{k}_i$ denotes the i-th row of $\boldsymbol{K}$. For $z = f'(\boldsymbol{y})$, the m-variable p-ary function f' has $|\mathcal{A}_{f'}(z)|$ preimages that we denote $(y_1^{(j)}, \ldots, y_m^{(j)})$, for $j \in [1, |\mathcal{A}_{f'}(z)|]$. From an image z, the adversary obtains that a linear system of m equations, namely $\{\boldsymbol{k}_i \boxdot \boldsymbol{r} = y_i^{(j)}$, for $i \in [1, m]\}$, is correct for one j. She can then collect $n+1$ samples $z_1, \ldots, z_{n+1}$ and solve the $\prod_{i=1}^{n+1} |\mathcal{A}_{f'}(z_i)|$ possible linear systems in $m\,(n+1)$ variables (in order to retrieve $\boldsymbol{K}$). Only one of these systems is the correct one, and the correct solution can be verified with more samples. Since by definition $|\mathcal{A}_{f'}(z_i)| \leq \mathsf{v}_{f'}$, the time complexity of this attack is upper bounded by $\mathcal{O}(\mathsf{v}_{f'}^{n+1} \cdot (m(n+1))^{\omega})$ (following the analysis of Section 4).

We extend this attack by considering algebraic systems of higher degree rather than linear systems. When z gives $|\mathcal{A}_{f'}(z)|$ preimages, instead of considering $m \cdot |\mathcal{A}_{f'}(z)|$ linear equations, we can build the m equations:

$$\prod_{j=1}^{|\mathcal{A}_{f'}(z)|} (\boldsymbol{k}_i \boxdot \boldsymbol{r} - y_i^{(j)}) = 0, \text{ for } i \in [1, m],$$

where the m equations have degree $|\mathcal{A}_{f'}(z)|$. We first observe that the cases we previously described, with $m\cdot|\mathcal{A}_{f'}(z)|$ equations of degree 1 and $m\cdot 1$ equations of degree $|\mathcal{A}_{f'}(z)|$, are two extreme cases. We can also create $m\cdot b$ equations of degree c such that $bc = |\mathcal{A}_{f'}(z)|$, where only one of the b system of equations is correct. As a result, the attack becomes a tradeoff between the number of polynomial systems of fixed degree and their algebraic degree, since it corresponds to the resolution of b^{n+1} polynomial systems of degree c, with $bc = |\mathcal{A}_{f'}(z)|$.

The complexity of the attack relies on two main factors: first, the size of the preimage sets considered $|\mathcal{A}_{f'}(z)|$, intervening in the number of systems and the degree of the polynomial systems; second, the probability to get samples z_i with small preimage sets. We argue that this attack is not a threat since either the time complexity is higher than the security level targeted when $\mathcal{A}_{f'}(z)$ is large, or the data complexity is too high when $\mathcal{A}_{f'}(z)$ is small. To do so, we first give the expression of $|\mathcal{A}_{f'}(z)|$ when f' is the direct sum of m p-ary functions. Then, we show how the complexity of the attack can be bounded from the values of $|\mathcal{A}_{f'}(z_i)|$ from the samples z_i an adversary obtains. Finally, we give a proposition to bound the probability of getting z_i such that the preimage set is small, and we conclude by applying these results to a practical example.

Concretely, when f' is the direct sum of m p-ary functions f_i, the expression of $|\mathcal{A}_{f'}(z)|$ is given by:

$$|\mathcal{A}_{f'}(z)| = \sum_{\substack{y_1,\dots,y_m\in\mathbb{F}_p\\ y_1+\cdots+y_m=z}} \prod_{i=1}^{m} |\mathcal{A}_{f_i}(y_i)|.$$

Then, since the time complexity of the attack comes from the resolution of b^{n+1} polynomial systems of degree c such that $bc = |\mathcal{A}_{f'}(z)|$, using the analysis of Section 4 the complexity is $\mathcal{O}(b^{n+1}\cdot(mV_c)^{\omega})$. We rewrite $b^{n+1}\cdot(mV_c)^{\omega}$ to express it relatively to $|\mathcal{A}_{f'}(z)|$ as follows:

$$\begin{aligned} b^{n+1}\cdot(mV_c)^{\omega} &= \left(\frac{|\mathcal{A}_{f'}(z)|}{c}\right)^{n+1}\binom{n+1+c}{n+1}^{\omega} m^{\omega},\\ &\geq \left(\frac{|\mathcal{A}_{f'}(z)|}{c}\right)^{n+1}\frac{c^{n+1}}{(n+1)!}\binom{n+1+c}{n+1}^{\omega-1} m^{\omega},\\ &\geq \frac{|\mathcal{A}_{f'}(z)|^{n+1}}{(n+1)!}. \end{aligned}$$

Finally, we use the following proposition in order to bound the probability of getting images coming from small preimage sets:

Proposition 4 (Probability of getting images from small preimage sets). *Let $f'\in\mathcal{F}_{p,m}$ and $B\in\mathbb{N}$ such that $B < p^m/|\mathrm{Im}(f')|$ where $\mathrm{Im}(\cdot)$ denotes the image, the probability $p_{f',B}$ of choosing at random $z\in\mathbb{F}_p^m$ such that $|\mathcal{A}_{f'}(z)|\leq B$ follows:*

$$p_{f',B} \leq \frac{(|\mathrm{Im}(f')|-1)B}{p^m}.$$

In particular, if f' is the direct sum of m p-ary functions with C_1^s and $mts < p$ (where $t = \lceil \log p \rceil$), the probability becomes:

$$p_{f',B} \leq \frac{mstB}{p^m}.$$

Proof. First, we show the result in the general case. By definition, the preimage sets $\mathcal{A}_{f'}(z)$ for $z \in \mathbb{F}_p$ are a partition of $\mathbb{F}_p^m$, Therefore:

$$p^m = \sum_{z \in \mathbb{F}_p} |\mathcal{A}_{f'}(z)| = \sum_{z \in \mathrm{Im}(f')} |\mathcal{A}_{f'}(z)|.$$

Since $B < \frac{p^m}{|\mathrm{Im}(f')|}$, at least one of the images is such that $|\mathcal{A}_{f'}(z)| > B$. Hence, at most $|\mathrm{Im}(f')| - 1$ images are such that $|\mathcal{A}_{f'}(z)| \leq B$, and the corresponding number of corresponding preimages is therefore at most $(|\mathrm{Im}(f')| - 1)B$, which allows us to conclude on $p_{f',B}$. For the particular case where f' is a direct sum of m p-ary functions from C_1^s, since $mst < p$ we have that $f(y) \in [0, st]$ since $f \in \mathsf{C}_1^s$ and $f'(\boldsymbol{y}) \in [0, mst]$, hence $|\mathrm{Im}(f')| \leq mst + 1$. □

Using the same parameters as in Section 5, namely $p = 2^{31} - 1$, $s = 2^{12}$, $n = 4$ and $m = 4$, we obtain that preimage sizes $|\mathcal{A}_{f'}(z)|$ larger than $\sqrt[5]{5!2^{124}} \approx 2^{26}$ lead to a time complexity larger than 2^{124}. We can then take the value $B = 2^{26}$ and apply Proposition 4, leading to a probability to get a sample z with at most B preimages lower than $\frac{4 \cdot 2^{12} \cdot 31 \cdot 2^{26}}{(2^{31}-1)^4)} \approx 2^{-79}$. Hence, our analysis shows that attacks exploiting worst-case leakages cannot succeed for adversaries with a data complexity bounded by 2^{79}, whereas side-channel security generally requires preventing attacks with data complexity in the 2^{50} range. As a result, we conclude that such attacks are not a concrete threat to the proposed instance.

6 Empirical validation

The previous sections showed that the LWPR assumption can remain secure in the generalized context of linear leakage functions. Yet, as illustrated in Figure 2, it is based on an adversarial strategy that interprets the leakages in $\mathbb{F}_p$ and security against adversaries following this strategy relies on two assumptions. Namely, it requires a bound on the degree of the leakage function and another bound on the quantization of the coefficients of this leakage function.

In the following, we therefore evaluate the validity of our assumptions based on a prototype hardware implementation (in Section 6.2 and Section 6.4) and discuss the applicability and relevance of our attack strategy (in Section 6.3). We conclude in Section 6.5, by confirming that the concrete leakage models we estimated from real measurements are covered by our security analysis. Beforehand, we describe the prototype implementation we considered in Section 6.1

	domain	codomain	degree	model coefficients
physical leakages	$\mathbb{F}_p^m$	$\mathbb{R}$	1 to δ	$\emptyset$
measured leakages	$\mathbb{F}_p^m$	$\{0, 1, \dots, s'\} \in \mathbb{Z}$	1 to δ	$\emptyset$
s-bounded leakages	$\mathbb{F}_p^m$	at most $\{0, 1, \dots, st\} \in \mathbb{F}_p$	1 (or more) *Sect. 6.2 (& 5)*	$\{0, 1, \dots, s\} \in \mathbb{F}_p$ *Sect. 6.4*

SCA model
Sect. 6.3

Figure 2: Adversarial strategy & leakage assumptions.

6.1 Prototype hardware implementation

We measured a parallel hardware implementation operating on 31-bit words running on a FPGA. The architecture of the computational core of our masked LWPR-based re-keying scheme is depicted in Figure 3. Concretely, the complete computation is performed by processing each share sequentially and the matrix multiplication for the share index d is split in $(N + 1) \times M = (4 + 1) \times 4 = 20$ modular multiplications over $\mathbb{F}_p$. More precisely, the core implements 4 modular multiplications in parallel (i.e., the results of the multiplications between the j-th column of a key share denoted $K^d_{*,0 \leq j < 5}$ and the j-th element of the nonce denoted r_j, where $r_4 = 1$). We used the efficient modular multiplication architecture based on DSPs proposed in [KSHS17] for this purpose. The results obtained for the multiplications are then added in 4 accumulators (i.e., one for each row) dedicated to the reconstruction of the resulting fresh key.

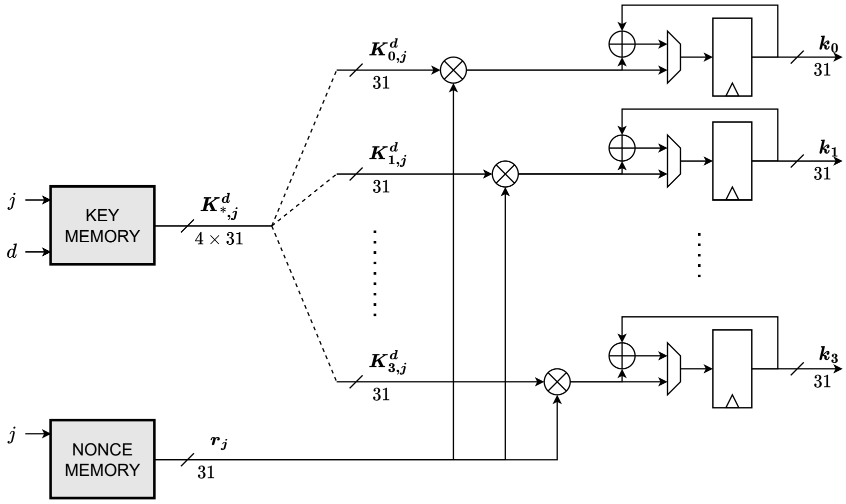

Figure 3: Masked LWPR-based re-keying: prototype hardware implementation.

As shown in Figure 4, the architecture of the key memory is designed to limit the impact of physical defaults on the implementation's security. In particular, the key shares are stored in independent physical memory units (i.e., BRAMs in the context of our FPGA implementation) and a register barrier is used before selecting the appropriate memory output. The latter is needed to avoid shares' recombinations that may be caused by glitches. Finally, a dedicated multiplexer at the input of the register barrier is used to drive the shares' values that are not currently processed to zero. Together with the fact that the shares are processed sequentially, this mechanism ensures the independence of the key shares.

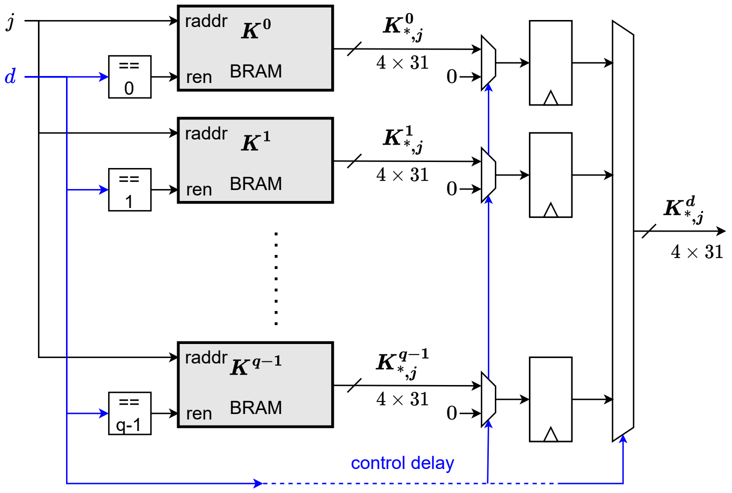

Figure 4: Key material memory architecture.

Overall, our prototype implementation matches the parallelism requirements of our security analysis since it recombines a full (124-bit) fresh key in a single clock cycle, the leakage of which we analyze in the rest of this section. Besides, we note that when analyzing the second (red) attack path described in the threat model of Figure 1, we are only interested in the value of the recombined fresh key.[4] Concretely, we therefore choose to measure an implementation with a single share (i.e., $q = 1$). This gives the most favorable leakages to the evaluator and therefore puts us as close as possible to the noise-free setting considered in our analyzes. This is because the implementation with $q = 1$ is the smallest one and a limited use of resources in turn reduces the noise in the leakages.

Our measurements were performed on the Sakura-G evaluation board, which embeds a XC6SLX75-2CSG484C Spartan6 FPGA. The traces were collected

[4] The first attack path (leveraging the leakages of the shared multiplications) was analyzed in [DMMS21] and the arguments of this previous work apply similarly.

using a Picoscope 5244D with a Tektronix CT-1 current probe, following the guidelines given in [BUS21]. In order to take into account the effect of congestion that can take place during the place-and-route step of a dense design, we performed three different syntheses using the ISE14.7 design toolchain. The first one is unconstrained. The second one is constrained in order to achieve 95% of resources' utilization density (using a PBLOCK area constraint). The last one further constraints the design by (artificially) enforcing that higher-capacity routing elements are included in the path of some bits. We next refer to these different syntheses as unconstrained, constrained and amplified. The layout of the unconstrained and constrained syntheses are illustrated in Figure 5.

6.2 Bounded degree of the leakage function

The first (and main) physical assumption used to rule out algebraic attacks in our previous analysis is that the leakage function has a bounded degree. More precisely, we claim security for linear leakage functions in Sections 4 and 5 and will generalize to higher-order (e.g., quadratic) functions in Section 7.

We next study whether this assumption can be reasonably fulfilled in practice. As detailed in Section 2.2, regression-based models are a natural tool for this purpose, since they can be estimated for different, more or less complex, basis functions. A simple case, considered in the previous sections, is to take the bits of the target intermediate value as basis functions. But higher-degree models can be built, culminating with the exhaustive model which corresponds to a (worst-case) profiling where all the mean leakage values are exhaustively estimated [CRR02]. Our goal is to therefore show that a simple linear model does not lead to significant information losses compared to the exhaustive one.

For this purpose, we analyzed the 4 leakage functions f_i of our target implementation. We analyzed them independently to reduce the noise. They all gave similar results. For the exhaustive model, and since building 2^{31} templates is computationally intensive, we rather estimated a subset of n_a average leakage traces (with $n_a = 10,000$), and stored them in a vector of average leakage traces $\overline{\boldsymbol{l}} = [\overline{l^1}, \overline{l^2}, \ldots, \overline{l^{n_a}}]$. We then computed the vectors corresponding to our regression-based linear models for the three different syntheses, evaluated on the same values. In each case, we first estimated the 31-element vector of coefficients $\hat{\boldsymbol{a}}$ using $n_p = 10,000$ profiling traces (meaning $\approx \frac{10,000}{31}$ traces per coefficient, which was sufficient for good estimation given the low noise level of our measurements). As a result, we obtained model predictions:

$$\boldsymbol{m}^u_{\mathsf{r1}} = [\mathsf{M}^u_{\mathsf{r1}}(\boldsymbol{v}^1), \mathsf{M}^u_{\mathsf{r1}}(\boldsymbol{v}^2), \ldots, \mathsf{M}^u_{\mathsf{r1}}(\boldsymbol{v}^{n_a})],$$

for the unconstrained synthesis, and similarly $\boldsymbol{m}^c_{\mathsf{r1}}$ & $\boldsymbol{m}^a_{\mathsf{r1}}$ for the constrained and amplified syntheses. Viewing these vectors as the samples of random variables $\overline{L}$, M^u_{r1}, M^c_{r1} and M^a_{r1}, we finally computed the correlation between the true (averaged) leakages and the models: $\hat{\rho}(M^u_{\mathsf{r1}}, \overline{L})$, $\hat{\rho}(M^c_{\mathsf{r1}}, \overline{L})$ and $\hat{\rho}(M^a_{\mathsf{r1}}, \overline{L})$.

These correlation coefficients are reported in Figure 6 in function of the number of traces used to obtain the average traces $\overline{l^i}$. We additionally report the

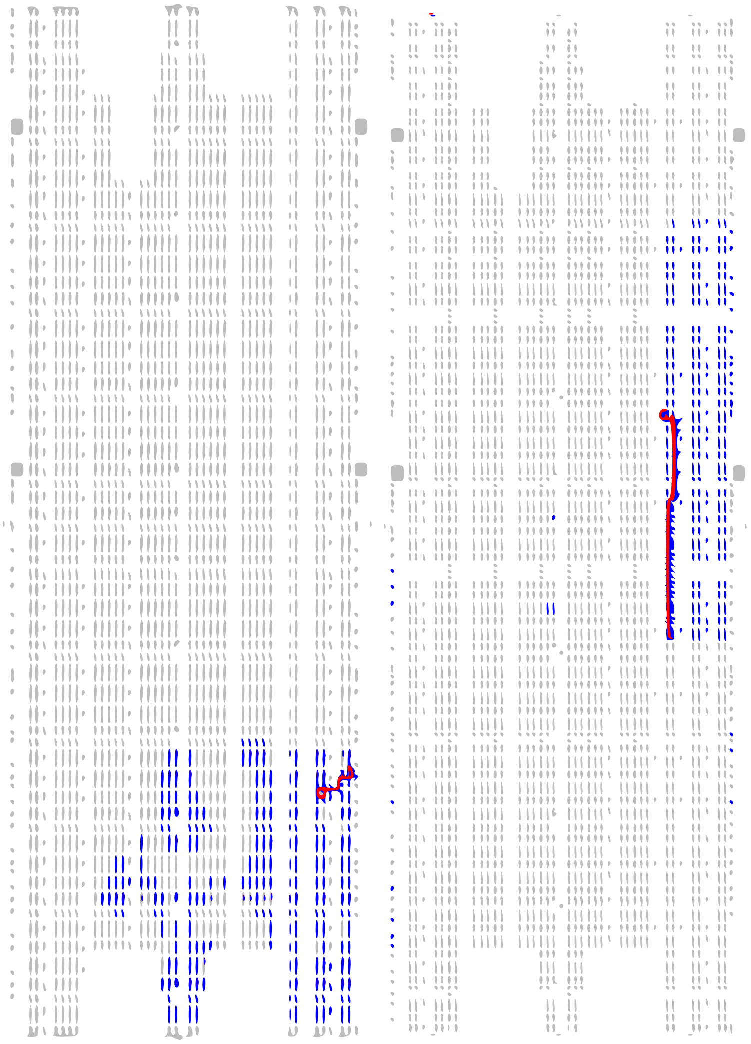

Figure 5: Layout of unconstrained (left) and constrained (right) syntheses. Unused resources are in white. Used resources are in blue. Routing is in red.

estimates obtained when correlating the real leakage vectors with a Hamming weight leakage model, and each curve in the figure is accompanied by a 95% bootstrap confidence interval. These results confirm that unconstrained syntheses (at the top of the figure) lead to high correlations already with the Hamming weight leakage model. By contrast, for constrained and amplified syntheses reflecting congestioned designs (at the middle and bottom of the figure) the correlation with the Hamming weight leakage model decreases while it remains close to one for the linear models. These experiments show that the generalization we propose indeed captures practically-relevant implementations. Admittedly, reaching such high correlation values does not preclude that higher-degree terms can be characterized and exploited. But it shows that most of the information leaked by our target implementation can be extracted with a simple linear model. So combined with (i) the generalization of Section 7 showing that security does not collapse as long as the leakage function remains sufficiently non-injective, and (ii) the fact that our modeling is conservative (since it assumes noise-free leakages), we conjecture that the bounded-degree assumption we require can be matched sufficiently well to provide practically secure LWPR instances.

6.3 Practical application of the attack

The attacks we study in this paper work by interpreting physical leakages as functions in $\mathbb{F}_p$. So far, we showed that such attacks are hard if the leakage function has a bounded degree and its coefficients have limited accuracy. We also assessed empirically that the bounded degree assumption can be reasonably matched in practice. We now describe how to mount these attacks thanks to

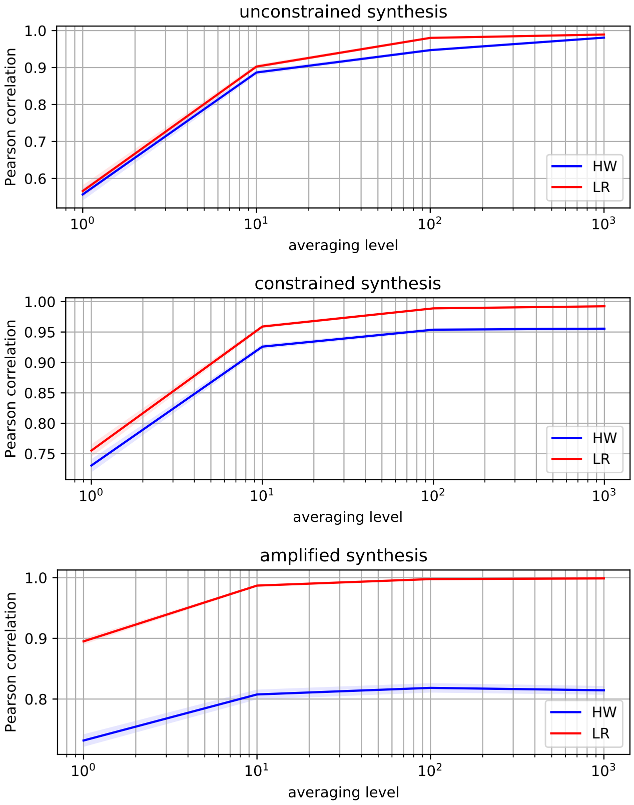

Figure 6: Correlation between real traces with different levels of averaging (on the X axis) and Hamming weight vs. regression-based linear models.

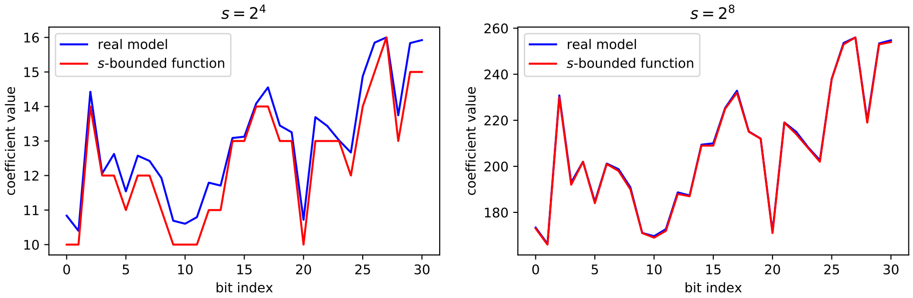

Figure 7: Discretization of the linear regression models (unconstrained synth.).

standard side-channel analysis tools, which will incidentally lead us to provide good indications that limiting the coefficient's accuracy is also reasonable.

Precisely, and as illustrated in Figure 2, we show how to move from measured leakages to s-bounded leakages thanks to the linear regression-based model of Section 2.2. For this purpose, we re-use the 31-element vectors of coefficients $\hat{\boldsymbol{a}}$ estimated using $n_p = 10,000$ profiling traces, for each of our three syntheses. We next observe that the value of Pearson's correlation is not affected if one of its variables is multiplied by a constant value. As a result, the following simple discretization process for the coefficients $\hat{\boldsymbol{a}}$ can be used:

$$\hat{\boldsymbol{a}}_{\mathsf{d}} = \left\lceil \hat{\boldsymbol{a}} \cdot \frac{s}{\mathsf{max}(\hat{\boldsymbol{a}})} \right\rceil .$$

We finally illustrate the coefficients of the linear models estimated from the unconstrained synthesis measurements together with the coefficients of the corresponding s-bounded functions in Figure 7. We can observe that already for $s = 2^8$ the matching between both is quite accurate. We further estimated the correlation $\hat{\rho}(M^s_{\mathsf{r}1}, M_{\mathsf{r}1})$ which was worth $1 - \phi$ with $\phi < 10^{-6}$ for $s = 2^8$.[5] We conclude that a quantization corresponding to $s = 2^{12}$ offers a sufficient granularity to discretize our linear leakage models nearly perfectly. In other words, while we cannot rule out that other strategies than interpreting the leakages in $\mathbb{F}_p$ as we consider can lead to better results, this section shows that if this is the best strategy, then the move from measured leakages to s-bounded ones does not imply a significant information loss for the values of s (e.g., 2^{12}) that our theoretical investigations tolerate. The results for the quantization of the models for moderately constrained and amplified syntheses are similar. We illustrate their respective coefficients in Figures 8 and 9 for completeness.

6.4 Bounded measurement accuracy

The previous sections suggested that using linear regression-based models is sufficient to accurately capture concretely-relevant leakage functions and that

[5] With a relative error of below 1%, which is easy to reach since the estimation is performed from modeled samples (rather than measured ones in Section 6.2).

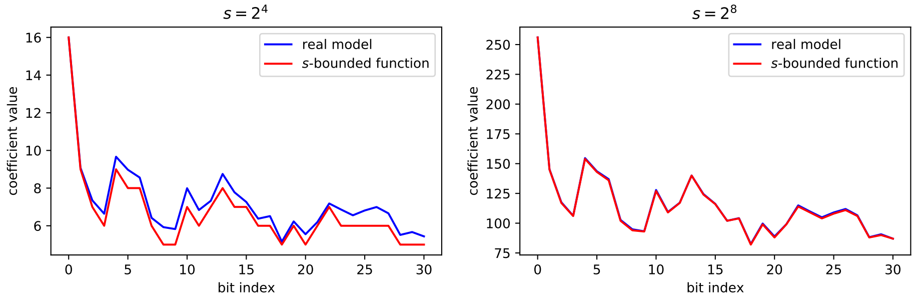

Figure 8: Discretization of the linear regression models (constrained synthesis).

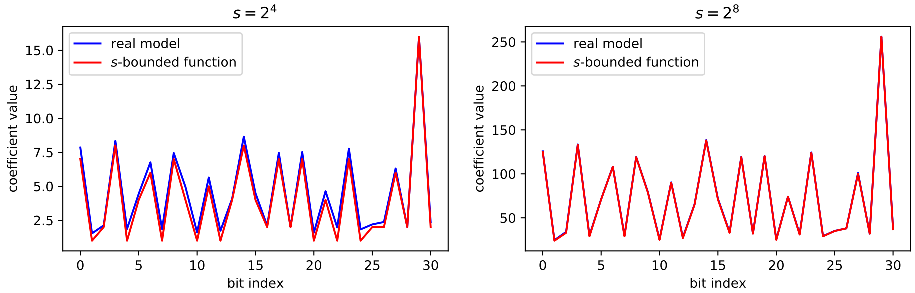

Figure 9: Discretization of the linear regression models (amplified synthesis).

increasing the coefficient accuracy of the s-bounded pseudo-linear functions beyond 2^{12} does not lead to significant improvements of their informativeness in our case study. Quite naturally, such assessments remain limited by their heuristic nature. For the analysis of the degree, this is unavoidable (since absence of evidence is not evidence of absence) and the only option to make it more robust is to consider higher-degree functions that could theoretically show up, which we will do in Section 7. But for the bounded coefficient accuracy, there is a complementary argument that could be given. For this purpose, the starting observation is that physical leakages are measured by converting an analog signal into a digital one. Concretely, the "vertical resolution" s' of modern oscilloscopes typically ranges from 8 to 12 bits, meaning that the measured leakage can take between 2^8 and 2^{12} values (possibly a bit more thanks to oversampling, at the cost of a reduced "horizontal" resolution). So if the increase of the s parameter from Definition 5 leads to a pseudo-linear function having a codomain with more than s' elements, it implies that the s-bounded leakage function is more informative than the measured one, which contradicts the data processing inequality. So for such leakage functions, the bounded coefficient accuracy can be directly connected to an assumption on the adversary's measurement apparatus.

However, increasing the s parameter does not always increase the cardinality of the leakage functions (e.g., think about the case where $t = 1$ for an increasing s). So one cannot directly bound the coefficient accuracy s based on the resolution of the oscilloscope s'. Yet, we note that most of the security claims in Sections 4 (resp., 5) depend on the product ts (resp., mts) which bounds the cardinality of the functions' co-domain and where t corresponds to the $\binom{t}{1}$ basis functions of a linear leakage model. So if the security analysis could be improved in order to rely only on this cardinality of the leakage functions, independent of the value of s (i.e., getting rid of the $ts < p$ and $mts < p$ conditions), we could replace the assumption on the bounded coefficient accuracy by an assumption on the resolution of the oscilloscope used to mount the attack. We leave the investigation of such an improved analysis as an interesting open problem.

6.5 Concrete security level

The three aforementioned implementations gave us three sets of real parameters, namely $\hat{\boldsymbol{a}}^u, \hat{\boldsymbol{a}}^c$ and $\hat{\boldsymbol{a}}^a$. Each of them can be discretized using the method of Section 6.3, resulting in three sets of $\mathbb{F}_p$ coefficients $\boldsymbol{a}_{\mathsf{d}}^u, \boldsymbol{a}_{\mathsf{d}}^c$ and $\boldsymbol{a}_{\mathsf{d}}^a$, themselves associated to an s-bounded pseudo-linear function $\mathsf{F}_{\boldsymbol{a}_{\mathsf{d}}^u}, \mathsf{F}_{\boldsymbol{a}_{\mathsf{d}}^c}$ and $\mathsf{F}_{\boldsymbol{a}_{\mathsf{d}}^a}$.

	$v_{\mathsf{F}_{\boldsymbol{a}_{\mathsf{d}}^*}}$	$w_{\mathsf{F}_{\boldsymbol{a}_{\mathsf{d}}^*}}$	$\mathsf{deg}(\mathsf{F}_{\boldsymbol{a}_{\mathsf{d}}^*})$	$\mathsf{nl}(\mathsf{F}_{\boldsymbol{a}_{\mathsf{d}}^*})$
Unconstrained	101748	1073657388	$\geq$ 101748	$\geq$ 2147381899
Constrained	213852	1073692455	$\geq$ 213852	$\geq$ 2147269795
Amplified	160593	1073709374	$\geq$ 160593	$\geq$ 2147323054

Table 1: $\mathsf{deg}(\mathsf{F}_{\boldsymbol{a}_{\mathsf{d}}^*})$ and $\mathsf{nl}(\mathsf{F}_{\boldsymbol{a}_{\mathsf{d}}^*})$ bounds of s-bounded pseudo-linear functions for the different implementations (computed using Proposition 3).

The results of Table 1 show the very high values of the degree and nonlinearity reached by these s-bounded pseudo-linear functions. As a result, algebraic attacks appear to be impracticable against the corresponding LWPR instances. The time complexities being several order of magnitude larger than the 124 bits of security we target, even more advanced approach (e.g., solving algebraic systems with Gröbner bases, using code-based techniques to approximate by smaller-degree functions) should not lead to successful attacks.

7 Further generalizations

The previous section confirmed the security that can be offered by concrete LWPR instances. It also showed that linear models are good abstractions to capture the features of actual leakage measurements. In this section, we show that the positive results obtained for s-bounded pseudo-linear functions extend naturally to higher-order leakage models, taking the quadratic case as an example. Despite not directly motivated by practice (as linear leakage models offered high level of correlation with real leakages in the previous section), we deem this result important to confirm that even in the hypothetical presence of such higher-order dependencies, the security of LWPR would not collapse.

Definition 7 (s-bounded pseudo-quadratic functions). *We denote $\mathsf{Q}_{\boldsymbol{a},\boldsymbol{b}}$ the function defined by a vector $\boldsymbol{a} \in \mathbb{F}_p^t$ and a stricly upper triangular matrix $\boldsymbol{b} \in \mathbb{F}_p^{t\times t}$, where $t = \lceil \log p \rceil$, as:*

$$\begin{aligned}\mathsf{Q}_{\boldsymbol{a},\boldsymbol{b}} :& \mathbb{F}_p \to \mathbb{F}_p \\ & y \mapsto \sum_{i=0}^{t-1} a_i \mathsf{g}(y)_i + \sum_{0\le i<j<t} b_{i(t+1)+j} \mathsf{g}(y)_i \mathsf{g}(y)_j .\end{aligned}$$

We call $\mathsf{Q}_{\boldsymbol{a},\boldsymbol{b}}$ s-bounded pseudo-quadratic when each $a_i, b_{i,j}$ belongs to $[0, s]$, with security parameter $s \in \mathbb{F}_p$ C_2^s the class of s-bounded pseudo-quadratic functions.

We give the following result for attacks exploiting any-case leakages.

Proposition 5 (Properties of s-bounded pseudo-quadratic functions). *Let $f \in \mathsf{C}_2^s$ with $\frac{t(t+1)}{2}s < p$ and $t = \lceil \log p \rceil$, then the following holds:*

- $\mathsf{v}_f \ge \left\lceil \frac{p}{st(t+1)/2+1} \right\rceil$,
- $\mathsf{w}_f \ge p - st(t+1)/2 - 1$.

And assuming $\mathsf{v}_f \neq p$, we further have:

- $\mathsf{deg}(f) \ge \left\lceil \frac{p}{st(t+1)/2+1} \right\rceil$,
- $\mathsf{nl}(f) \ge \min\left(p - \mathsf{v}_f, \max\left(\left\lceil \frac{p}{st(t+1)/2+1} \right\rceil - 1, p - st(t+1)/2 - 1\right)\right)$.

Proof. Since $\frac{t(t+1)}{2}s < p$, we obtain that for all $x \in \mathbb{F}_p$, $0 \le f(x) \le \frac{t(t+1)}{2}s$, hence $\mathsf{w}_f \ge p - \frac{t(t+1)}{2}s - 1$. Thereafter, $f(x)$ can take at most $\frac{t(t+1)}{2}s + 1$ values, hence there exists at least a preimage set with cardinal $\left\lceil \frac{p}{st(t+1)/2+1} \right\rceil$ or more. Finally, the last items are obtained by using Proposition 1 and Proposition 2 respectively, combined with the two first items. □

Note that the condition $ts < p$ which avoids the reduction when computing the leakage function in the analysis of the linear case is replaced by a condition $\frac{t(t+1)}{2}s < p$, where $\frac{t(t+1)}{2} = \binom{t}{1} + \binom{t}{2}$ is the number of basis functions in the quadratic model. This last result shows that s-bounded pseudo-quadratic functions behave similarly to s-bounded pseudo-linear functions. The only difference is that the security parameter s is a bit more constrained, so that $\frac{t(t+1)}{2}s < p$. Yet, with $p = 2^{31} - 1$ and $t = \lceil \log p \rceil$, security remains guaranteed for $s = 2^{12}$ (with a large security margin). In a similar manner, it could be shown that the estimated cost of algebraic attacks against LWPR is maintained for larger degrees, if the leakage function it relies on remains sufficiently non-injective, with a constraint on s evolving with the number of basis functions in the model.

As for attacks using worst-case leakages as studied in Section 5, the condition is slightly more restricted since we need $m\frac{t(t+1)}{2}s < p$ in order to apply Proposition 4. It nevertheless remains sufficient in the quadratic case. Besides, it could again be that an improved analysis leads to relexed conditions (or tighter bounds), which we leave as an interesting scope for further research.

8 Conclusions and open problems

By showing that the LWPR assumption remains secure for a wide class of practically-relevant leakage functions, this work makes an important step in improving its usability. As a result, it strengthens the motivation for using re-keying schemes operating in prime fields, since they provide significant security improvements over their binary counterparts. Concretely, our investigations suggest that parallel instances of LWPR have good potential to be secure. The generalization of Section 7 further shows that increasing the degree of the leakage function is not a direct threat, so that the main assumption for LWPR to be hard is the quite natural requirement that the leakages are sufficiently non-injective.

Our results open several interesting avenues for further research. First, Section 3 recalled the challenge of securing serial LWPR implementations against leakage and suggested shuffling as a possible option for this purpose. Its concrete investigation is therefore a natural next step. Second, our analyzes could be improved in order to rely only on the degree of the leakage functions and the cardinality of their co-domain. As discussed in Section 6.4, this could lead to formally connect the bouned coefficient accuracy hypothesis with the resolution of the adversary's measurement apparatus. Third, we for now consider univariate leakages. They reasonably match the hardware implementation context we evaluated, where it is possible to implement the un-masking of the ephemeral key in a single cycle. But multivariate generalizations are an important scope for further investigations and will be especially relevant in a software implementation context where (for example) the reduction that has to take place after the un-masking may take several cycles. Dealing with more leaky software implementations (even with shuffling) may also motivate stepping back to a less conservative model, where security is argued based on the difficulty for the adversary to infer leakage values without errors, possibly formalized by a Learning With Physical Rounding and Errors problem. LWPR being an aggressive and new physical assumption, improving the understanding of the best cryptanalysis techniques to break it naturally remains needed as well. For example, our current attacks do not exploit key guessing strategies (which should at least imply a slight increase of the security parameters). Eventually, it would be worth studying the conditions upon which hard physical learning problems can be connected and reduced to standard (mathematical) hard learning problems.

Acknowledgments. Pierrick Méaux was supported by the ERC project 787390 (acronym CLOUDMAP). François-Xavier Standaert is a senior research associate of the Belgian Fund for Scientific Research (FNRS-F.R.S.). This work has been funded in parts by the European Union through the ERC project 724725 (acronym SWORD) and by the Walloon Region Win2Wal project PIRATE.

References

BCF$^+$15. Sonia Belaïd, Jean-Sébastien Coron, Pierre-Alain Fouque, Benoît Gérard, Jean-Gabriel Kammerer, and Emmanuel Prouff. Improved side-channel analysis of finite-field multiplication. In *CHES*, volume 9293 of *Lecture Notes in Computer Science*, pages 395–415. Springer, 2015.

BCO04. Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In *CHES*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.

BDF$^+$17. Gilles Barthe, François Dupressoir, Sebastian Faust, Benjamin Grégoire, François-Xavier Standaert, and Pierre-Yves Strub. Parallel implementations of masking schemes and the bounded moment leakage model. In *EUROCRYPT (1)*, volume 10210 of *Lecture Notes in Computer Science*, pages 535–566, 2017.

BDK$^+$18. Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - kyber: A cca-secure module-lattice-based KEM. In *EuroS&P*, pages 353–367. IEEE, 2018.

BFG14. Sonia Belaïd, Pierre-Alain Fouque, and Benoît Gérard. Side-channel analysis of multiplications in GF(2128) - application to AES-GCM. In *ASIACRYPT (2)*, volume 8874 of *Lecture Notes in Computer Science*, pages 306–325. Springer, 2014.

BFS15. Magali Bardet, Jean-Charles Faugère, and Bruno Salvy. On the complexity of the F5 gröbner basis algorithm. *J. Symb. Comput.*, 70:49–70, 2015.

BGL$^+$14. Hai Brenner, Lubos Gaspar, Gaëtan Leurent, Alon Rosen, and François-Xavier Standaert. FPGA implementations of SPRING - and their countermeasures against side-channel attacks. In *CHES*, volume 8731 of *Lecture Notes in Computer Science*, pages 414–432. Springer, 2014.

BIP$^+$18. Dan Boneh, Yuval Ishai, Alain Passelègue, Amit Sahai, and David J. Wu. Exploring crypto dark matter: - new simple PRF candidates and their applications. In *TCC (2)*, volume 11240 of *Lecture Notes in Computer Science*, pages 699–729. Springer, 2018.

BPR12. Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In *EUROCRYPT*, volume 7237 of *Lecture Notes in Computer Science*, pages 719–737. Springer, 2012.

BUS21. Davide Bellizia, Balazs Udvarhelyi, and François-Xavier Standaert. Towards a better understanding of side-channel analysis measurements setups. In *CARDIS*, volume 13173 of *Lecture Notes in Computer Science*, pages 64–79. Springer, 2021.

BV11. Zvika Brakerski and Vinod Vaikuntanathan. Fully homomorphic encryption from ring-lwe and security for key dependent messages. In *CRYPTO*, volume 6841 of *Lecture Notes in Computer Science*, pages 505–524. Springer, 2011.

Car21. Claude Carlet. *Boolean Functions for Cryptography and Coding Theory.* Cambridge University Press, 2021.

CGP$^+$12. Jean-Sébastien Coron, Christophe Giraud, Emmanuel Prouff, Soline Renner, Matthieu Rivain, and Praveen Kumar Vadnala. Conversion of security proofs from one leakage model to another: A new issue. In *COSADE*, volume 7275 of *Lecture Notes in Computer Science*, pages 69–81. Springer, 2012.

CHKP10. David Cash, Dennis Hofheinz, Eike Kiltz, and Chris Peikert. Bonsai trees, or how to delegate a lattice basis. In *EUROCRYPT*, volume 6110 of *Lecture Notes in Computer Science*, pages 523–552. Springer, 2010.

CHMS22. Orel Cosseron, Clément Hoffmann, Pierrick Méaux, and François-Xavier Standaert. Towards case-optimized hybrid homomorphic encryption - featuring the elisabeth stream cipher. In *ASIACRYPT (3)*, volume 13793 of *Lecture Notes in Computer Science*, pages 32–67. Springer, 2022.

Cou02. Nicolas T. Courtois. Higher order correlation attacks, XL algorithm and cryptanalysis of toyocrypt. In Pil Joong Lee and Chae Hoon Lim, editors, *Information Security and Cryptology - ICISC 2002, 5th International Conference Seoul, Korea, November 28-29, 2002, Revised Papers*, volume 2587 of *Lecture Notes in Computer Science*, pages 182–199. Springer, 2002.

CRR02. Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. Template attacks. In *CHES*, volume 2523 of *Lecture Notes in Computer Science*, pages 13–28. Springer, 2002.

CS20. Gaëtan Cassiers and François-Xavier Standaert. Trivially and efficiently composing masked gadgets with probe isolating non-interference. *IEEE Trans. Inf. Forensics Secur.*, 15:2542–2555, 2020.

CS21. Gaëtan Cassiers and François-Xavier Standaert. Provably secure hardware masking in the transition- and glitch-robust probing model: Better safe than sorry. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(2):136–158, 2021.

DFH$^+$16. Stefan Dziembowski, Sebastian Faust, Gottfried Herold, Anthony Journault, Daniel Masny, and François-Xavier Standaert. Towards sound fresh re-keying with hard (physical) learning problems. In *CRYPTO (2)*, volume 9815 of *Lecture Notes in Computer Science*, pages 272–301. Springer, 2016.

DGH$^+$21. Itai Dinur, Steven Goldfeder, Tzipora Halevi, Yuval Ishai, Mahimna Kelkar, Vivek Sharma, and Greg Zaverucha. Mpc-friendly symmetric cryptography from alternating moduli: Candidates, protocols, and applications. In *CRYPTO (4)*, volume 12828 of *Lecture Notes in Computer Science*, pages 517–547. Springer, 2021.

DKL$^+$18. Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. Crystals-dilithium: A lattice-based digital signature scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):238–268, 2018.

DMMS21. Sébastien Duval, Pierrick Méaux, Charles Momin, and François-Xavier Standaert. Exploring crypto-physical dark matter and learning with physical rounding towards secure and efficient fresh re-keying. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(1):373–401, 2021.

Fau99. Jean-Charles Faugère. A new efficient algorithm for computing Groebner bases. *Journal of Pure and Applied Algebra*, pages 61–88, june 1999.

Fau02. Jean Charles Faugère. A new efficient algorithm for computing gröbner bases without reduction to zero (f5). In *Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation, ISSAC 2002*, page 75–83, 2002.

Gen09. Craig Gentry. Fully homomorphic encryption using ideal lattices. In *STOC*, pages 169–178. ACM, 2009.

GJ19. Qian Guo and Thomas Johansson. A new birthday-type algorithm for attacking the fresh re-keying countermeasure. *Inf. Process. Lett.*, 146:30–34, 2019.

GLP06. Benedikt Gierlichs, Kerstin Lemke-Rust, and Christof Paar. Templates vs. stochastic methods. In *CHES*, volume 4249 of *Lecture Notes in Computer Science*, pages 15–29. Springer, 2006.

GLS14. Lubos Gaspar, Gaëtan Leurent, and François-Xavier Standaert. Hardware implementation and side-channel analysis of lapin. In *CT-RSA*, volume 8366 of *Lecture Notes in Computer Science*, pages 206–226. Springer, 2014.

GPV08. Craig Gentry, Chris Peikert, and Vinod Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, pages 197–206. ACM, 2008.

GR17. Dahmun Goudarzi and Matthieu Rivain. How fast can higher-order masking be in software? In *EUROCRYPT (1)*, volume 10210 of *Lecture Notes in Computer Science*, pages 567–597, 2017.

HB01. Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. In *ASIACRYPT*, volume 2248 of *Lecture Notes in Computer Science*, pages 52–66. Springer, 2001.

HKL$^+$12. Stefan Heyse, Eike Kiltz, Vadim Lyubashevsky, Christof Paar, and Krzysztof Pietrzak. Lapin: An efficient authentication protocol based on ring-lpn. In *FSE*, volume 7549 of *Lecture Notes in Computer Science*, pages 346–365. Springer, 2012.

HLM$^+$23. Clément Hoffmann, Benoît Libert, Charles Momin, Thomas Peters, and François-Xavier Standaert. POLKA: towards leakage-resistant post-quantum cca-secure public key encryption. In *Public Key Cryptography (1)*, volume 13940 of *Lecture Notes in Computer Science*, pages 114–144. Springer, 2023.

HOM06. Christoph Herbst, Elisabeth Oswald, and Stefan Mangard. An AES smart card implementation resistant to power analysis attacks. In *ACNS*, volume 3989 of *Lecture Notes in Computer Science*, pages 239–252, 2006.

Hou18. Xiang-dong Hou. *Lectures on Finite Fields*, volume 190. American Mathematical Society, 2018.

ISW03. Yuval Ishai, Amit Sahai, and David A. Wagner. Private circuits: Securing hardware against probing attacks. In *CRYPTO*, volume 2729 of *Lecture Notes in Computer Science*, pages 463–481. Springer, 2003.

KSHS17. Philipp Koppermann, Fabrizio De Santis, Johann Heyszl, and Georg Sigl. Automatic generation of high-performance modular multipliers for arbitrary mersenne primes on fpgas. In *HOST*, pages 35–40. IEEE Computer Society, 2017.

Man04. Stefan Mangard. Hardware countermeasures against DPA ? A statistical analysis of their effectiveness. In *CT-RSA*, volume 2964 of *Lecture Notes in Computer Science*, pages 222–235. Springer, 2004.

MOP07. Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007.

MPG05. Stefan Mangard, Thomas Popp, and Berndt M. Gammel. Side-channel leakage of masked CMOS gates. In *CT-RSA*, volume 3376 of *Lecture Notes in Computer Science*, pages 351–365. Springer, 2005.

MSGR10. Marcel Medwed, François-Xavier Standaert, Johann Großschädl, and Francesco Regazzoni. Fresh re-keying: Security against side-channel and fault attacks for low-cost devices. In *AFRICACRYPT*, volume 6055 of *Lecture Notes in Computer Science*, pages 279–296. Springer, 2010.

NDGJ21. Kalle Ngo, Elena Dubrova, Qian Guo, and Thomas Johansson. A side-channel attack on a masked IND-CCA secure saber KEM implementation. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2021(4):676–707, 2021.

Pie12. Krzysztof Pietrzak. Cryptography from learning parity with noise. In *SOFSEM*, volume 7147 of *Lecture Notes in Computer Science*, pages 99–114. Springer, 2012.

PM16. Peter Pessl and Stefan Mangard. Enhancing side-channel analysis of binary-field multiplication with bit reliability. In *CT-RSA*, volume 9610 of *Lecture Notes in Computer Science*, pages 255–270. Springer, 2016.

Reg05. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *STOC*, pages 84–93. ACM, 2005.

Reg10. Oded Regev. The learning with errors problem (invited survey). In *Computational Complexity Conference*, pages 191–204. IEEE Computer Society, 2010.

RRCB20. Prasanna Ravi, Sujoy Sinha Roy, Anupam Chattopadhyay, and Shivam Bhasin. Generic side-channel attacks on cca-secure lattice-based PKE and kems. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2020(3):307–335, 2020.

SLP05. Werner Schindler, Kerstin Lemke, and Christof Paar. A stochastic model for differential side channel cryptanalysis. In *CHES*, volume 3659 of *Lecture Notes in Computer Science*, pages 30–46. Springer, 2005.

SMY09. François-Xavier Standaert, Tal Malkin, and Moti Yung. A unified framework for the analysis of side-channel key recovery attacks. In *EUROCRYPT*, volume 5479 of *Lecture Notes in Computer Science*, pages 443–461. Springer, 2009.

SPRQ06. François-Xavier Standaert, Eric Peeters, Gaël Rouvroy, and Jean-Jacques Quisquater. An overview of power analysis attacks against field programmable gate arrays. *Proc. IEEE*, 94(2):383–394, 2006.

UBS21. Balazs Udvarhelyi, Olivier Bronchain, and François-Xavier Standaert. Security analysis of deterministic re-keying with masking and shuffling: Application to ISAP. In *COSADE*, volume 12910 of *Lecture Notes in Computer Science*, pages 168–183. Springer, 2021.

UXT$^+$22. Rei Ueno, Keita Xagawa, Yutaro Tanaka, Akira Ito, Junko Takahashi, and Naofumi Homma. Curse of re-encryption: A generic power/em analysis on post-quantum kems. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2022(1):296–322, 2022.

VMKS12. Nicolas Veyrat-Charvillon, Marcel Medwed, Stéphanie Kerckhof, and François-Xavier Standaert. Shuffling against side-channel attacks: A comprehensive study with cautionary note. In *ASIACRYPT*, volume 7658 of *Lecture Notes in Computer Science*, pages 740–757. Springer, 2012.

YC04. Bo-Yin Yang and Jiun-Ming Chen. All in the XL family: Theory and practice. In Choonsik Park and Seongtaek Chee, editors, *Information Security and Cryptology - ICISC 2004, 7th International Conference, Seoul, Korea, December 2-3, 2004, Revised Selected Papers*, volume 3506 of *Lecture Notes in Computer Science*, pages 67–86. Springer, 2004.

YS16. Yu Yu and John P. Steinberger. Pseudorandom functions in almost constant depth from low-noise LPN. In *EUROCRYPT (2)*, volume 9666 of *Lecture Notes in Computer Science*, pages 154–183. Springer, 2016.